

Dbvisit - How to solve Archivelog Gap on Standby

Olá, nesse artigo iremos aprender como automatizar a detecção e correção de Archive Gaps em ambientes Oracle Data Standby utilizando **RMAN** e **Dbvisit Standby**.

O que é um Archive Gap?

Um **Archive Gap** ocorre quando um ou mais arquivos de archive log necessários para recuperação do banco de dados standby não estão disponíveis no destino.

Esse problema pode ocorrer por diversos motivos:

- Falha de rede
- Espaço insuficiente em disco
- Interrupção do Dbvisit
- Exclusão acidental de archives
- Problemas operacionais

Quando isso acontece, o banco de dados standby não consegue continuar a aplicação dos redo logs, ficando defasado em relação ao ambiente primário.

Como identificar um Archive Gap?

O Dbvisit fornece um relatório de status através do comando:

```
dbvcli -d cylla -t oracle lgr --no-prompt
```

Exemplo de saída:

```
Current source sequence           : 4094
Archived source sequence          : 4093
Next required destination recovery sequence: 4085
Archive log gap                   : 9
```

Nesse cenário:

- O banco primário está no sequence 4094
- O standby necessita do sequence 4085
- Existem 9 archives pendentes

Como resolver manualmente?

Tradicionalmente o DBA precisa:

- Identificar o gap
- Localizar os archives faltantes
- Restaurar os archives através do RMAN
- Reenviar os logs usando o Dbvisit
- Aplicar os logs no standby

- Validar a sincronização

Em ambientes produtivos essa atividade pode consumir bastante tempo e está sujeita a falhas operacionais.

Automatizando o processo

Para eliminar a necessidade de intervenção manual foi desenvolvido um script Shell responsável por:

- Verificar se a instância local é PRIMARY
- Identificar Archive Gaps automaticamente
- Restaurar archives através do RMAN
- Reenviar os logs usando Dbvisit
- Aplicar os logs no standby
- Validar o resultado final

Verificando o Database Role

Por segurança, o script executa somente em bancos de dados PRIMARY.

A validação é realizada através da view:

```
SELECT DATABASE_ROLE  
FROM V$DATABASE;
```

Resultado esperado:

```
PRIMARY
```

Caso o banco esteja configurado como:

```
PHYSICAL STANDBY
```

o script é encerrado automaticamente.

Detecção automática do Gap

O script utiliza o comando:

```
dbvcli -d "$CONFIG" -t oracle lgr --no-prompt
```

Em seguida são extraídas as seguintes informações:

Campo	Descrição
Current Source Sequence	Último sequence gerado pelo primário
Archived Source Sequence	Último archive enviado
Next Required Recovery Sequence	Próximo sequence necessário no standby
Archive Log Gap	Quantidade de archives faltantes

Restauração automática com RMAN

Quando um gap é identificado, o script gera dinamicamente um arquivo de comandos RMAN.

Exemplo:

```
RUN {  
  RESTORE ARCHIVELOG  
  FROM SEQUENCE 4085  
  UNTIL SEQUENCE 4094;  
}
```

A execução é realizada através de:

```
rman target / cmdfile=restore_arch_cylla.rman
```

Reenvio dos Archives

Após a restauração dos arquivos, o Dbvisit é utilizado para reenviar os logs ao standby.

```
dbvcli -d "$CONFIG" -t oracle send-logs --no-prompt
```

Aplicação dos Archives

Em seguida o processo de recuperação é iniciado.

```
dbvcli -d "$CONFIG" -t oracle apply-logs --no-prompt
```

Esse passo garante que o standby processe os archives restaurados.

Validação Final

Ao término da execução o script realiza uma nova verificação:

```
dbvcli -d "$CONFIG" -t oracle lgr --no-prompt
```

Resultado esperado:

```
Archive log gap : 0
```

Isso confirma que o ambiente voltou a ficar sincronizado.

Benefícios da solução

A implementação trouxe diversas vantagens operacionais:

- Redução de intervenção manual
- Menor tempo de indisponibilidade do standby
- Recuperação automática de gaps
- Maior confiabilidade operacional
- Padronização do processo
- Facilidade de auditoria

Além disso, o script registra:

- Data de início

- Data de término
- Status da execução
- Sequências restauradas
- Resultado final

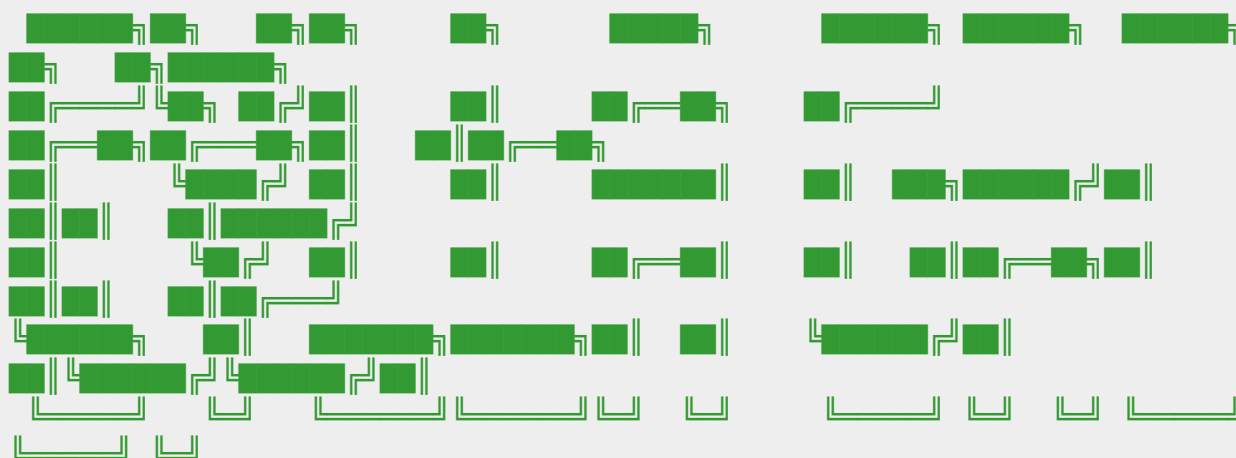
Segue abaixo o script completo:

[dbvisit_check_gap.sh](#)

```
#!/bin/bash
```

```
clear
```

```
cat <<'EOF'
```



DBVISIT GAP AUTO-RECOVERY TOOL

Author : Leonardo Lopes
Position : CEO
Company : Cylla Group
Website : <https://cylla.group>
Blog : <https://blog.dbahero.com>
Version : 1.0

Purpose:

Automatic detection and recovery of Archive Log Gaps
using Oracle RMAN and Dbvisit Standby.

```
EOF
```

```
echo
```

```
CONFIG="cylla"  
ORACLE_SID="cylla"  
export ORACLE_HOME="/u01/app/oracle/product/19.0.0/dbhome_1"  
export DBVISIT_HOME=/usr/dbvisit/standbyp  
export PATH=$ORACLE_HOME/bin:$PATH:$DBVISIT_HOME/bin
```

```

ORAENV_ASK=NO
. oraenv >> /dev/null

START_TIME=$(date '+%Y-%m-%d %H:%M:%S')

echo "=====
echo "Dbvisit Gap Auto-Recovery"
echo "Started at           : $START_TIME"
echo "Configuration       : $CONFIG"
echo "Oracle SID           : $ORACLE_SID"
echo "=====

DB_ROLE=$(sqlplus -s / as sysdba <<EOF
set heading off feedback off verify off echo off pagesize 0
select database_role from v\$database;
exit
EOF
)

DB_ROLE=$(echo "$DB_ROLE" | tr -d '[:space:]')

echo "Database role       : $DB_ROLE"

if [ "$DB_ROLE" != "PRIMARY" ]; then
    echo "Status           : Database is not PRIMARY. Skipping
execution."

    END_TIME=$(date '+%Y-%m-%d %H:%M:%S')
    echo "Finished at       : $END_TIME"
    echo "=====
    exit 0
fi

OUTPUT=$(dbvcli -d "$CONFIG" -t oracle lgr --no-prompt)

CURRENT_SEQ=$(echo "$OUTPUT" | awk -F: '/Current source sequence/
{print $2}' | tr -cd '0-9')
ARCHIVED_SEQ=$(echo "$OUTPUT" | awk -F: '/Archived source sequence/
{print $2}' | tr -cd '0-9')
NEXT_REQUIRED_SEQ=$(echo "$OUTPUT" | awk -F: '/Next required
destination recovery sequence/ {print $2}' | tr -cd '0-9')
ARCHIVE_GAP=$(echo "$OUTPUT" | awk -F: '/Archive log gap/ {print $2}' |
tr -cd '0-9')

echo "Current sequence     : $CURRENT_SEQ"
echo "Archived sequence     : $ARCHIVED_SEQ"
echo "Next required seq     : $NEXT_REQUIRED_SEQ"
echo "Archive gap           : $ARCHIVE_GAP"

if [ -z "$ARCHIVE_GAP" ] || [ "$ARCHIVE_GAP" -eq 0 ]; then
    echo "Status           : No archive gap detected."

```

```

    END_TIME=$(date '+%Y-%m-%d %H:%M:%S')
    echo "Finished at           : $END_TIME"
    echo "===== "
    exit 0
fi

FROM_SEQ="$NEXT_REQUIRED_SEQ"
TO_SEQ="$CURRENT_SEQ"

if ! [[ "$FROM_SEQ" =~ ^[0-9]+$ ]] || ! [[ "$TO_SEQ" =~ ^[0-9]+$ ]];
then
    echo "ERROR                : Invalid sequence values."
    echo "FROM_SEQ              : [$FROM_SEQ]"
    echo "TO_SEQ                 : [$TO_SEQ]"

    END_TIME=$(date '+%Y-%m-%d %H:%M:%S')
    echo "Finished at           : $END_TIME"
    echo "===== "
    exit 1
fi

echo "Missing sequences      : $FROM_SEQ to $TO_SEQ"
echo "Action                  : Restoring archivelogs from RMAN backup..."

RMAN_CMD="/tmp/restore_arch_${CONFIG}.rman"

cat > "$RMAN_CMD" <<EOF
RUN {
    RESTORE ARCHIVELOG FROM SEQUENCE ${FROM_SEQ} UNTIL SEQUENCE
    ${TO_SEQ};
}
EOF

echo "RMAN command file      : $RMAN_CMD"
cat "$RMAN_CMD"

rman target / cmdfile="$RMAN_CMD"

if [ $? -ne 0 ]; then
    echo "ERROR                : RMAN restore failed."

    END_TIME=$(date '+%Y-%m-%d %H:%M:%S')
    echo "Finished at           : $END_TIME"
    echo "===== "
    exit 1
fi

echo "Status                  : RMAN restore completed."

echo "Action                  : Sending restored logs with Dbvisit..."

```

```

dbvcli -d "$CONFIG" -t oracle send-logs --no-prompt

if [ $? -ne 0 ]; then
    echo "ERROR                      : Dbvisit send-logs failed."

    END_TIME=$(date '+%Y-%m-%d %H:%M:%S')
    echo "Finished at                : $END_TIME"
    echo "===== "
    exit 1
fi

echo "Status                      : Dbvisit send-logs completed."

echo "Action                      : Applying logs on standby..."
dbvcli -d "$CONFIG" -t oracle apply-logs --no-prompt

if [ $? -ne 0 ]; then
    echo "ERROR                      : Dbvisit apply-logs failed."

    END_TIME=$(date '+%Y-%m-%d %H:%M:%S')
    echo "Finished at                : $END_TIME"
    echo "===== "
    exit 1
fi

echo "Status                      : Dbvisit apply-logs completed."

echo "Final log gap report:"
dbvcli -d "$CONFIG" -t oracle lgr --no-prompt

END_TIME=$(date '+%Y-%m-%d %H:%M:%S')

echo "===== "
echo "Finished at                : $END_TIME"
echo "Status                      : Completed successfully."
echo "===== "

exit 0

```

Conclusão

Automatizar atividades operacionais é uma das melhores formas de aumentar a confiabilidade de ambientes Oracle.

A combinação entre **RMAN** e **Dbvisit Standby** permitiu construir uma solução simples, eficiente e capaz de recuperar automaticamente gaps de archives sem intervenção humana.

Em ambientes críticos essa abordagem reduz significativamente o tempo de recuperação e garante que o banco standby permaneça sincronizado com o ambiente de produção.

- Até a próxima.

Dbvisit - How to solve Archivelog Gap on Standby

— Autor: [Leonardo Lopes](#) 15/06/2026 05H:32

From:

<https://blog.dbahero.com/> -

Permanent link:

<https://blog.dbahero.com/doku.php?id=dbvisitgapscript>

Last update: **15/06/2026 05H:32**

